

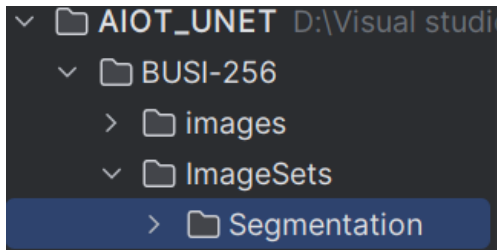
# A 档模型 UNet

--by 2213409 石彬辰

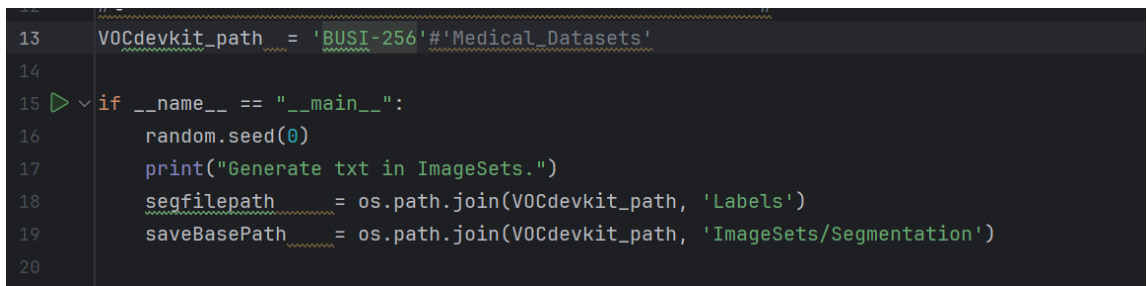
## BUSI-256 数据集

### 训练步骤

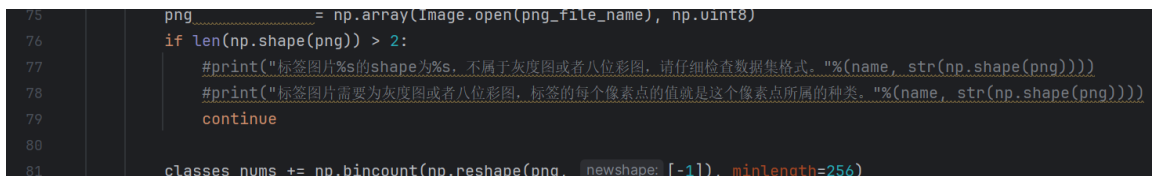
将数据集 BUSI-256 拷贝到项目根目录下,改变文件结构,images->images,masks->labels,新建文件夹 ImageSets/Segmentation



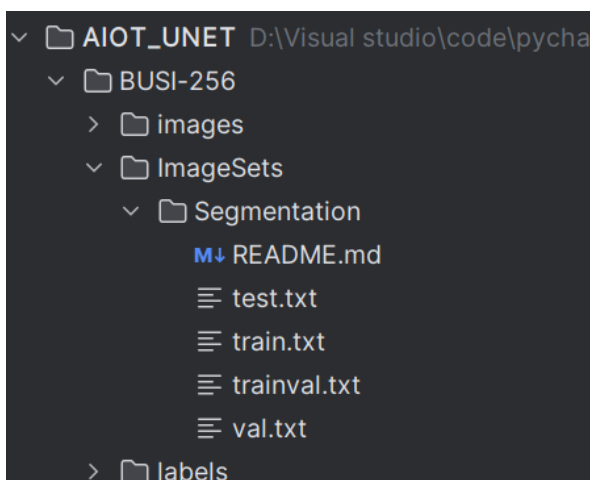
如下修改 voc\_annotation.py



由于部分 labels 中的遮罩图片为 24 位,修改 voc\_annotation.py 如下部分,跳过这些图片



使用 voc\_annotation.py 生成对应的 txt 文件



修改 train\_medical.py 中的数据集路径,添加环境变量语句

```

214 # 数据集路径
215 #-----#
216 VOCdevkit_path = 'BUSI-256'#'Medical_Datasets'
217 #-----#
218 # 建议选项:

```

```

116 input_shape = [256, 256]
117

```

```

20 os.environ["KMP_DUPLICATE_LIB_OK"]="TRUE"

```

运行 tarin\_medical.py 开始训练

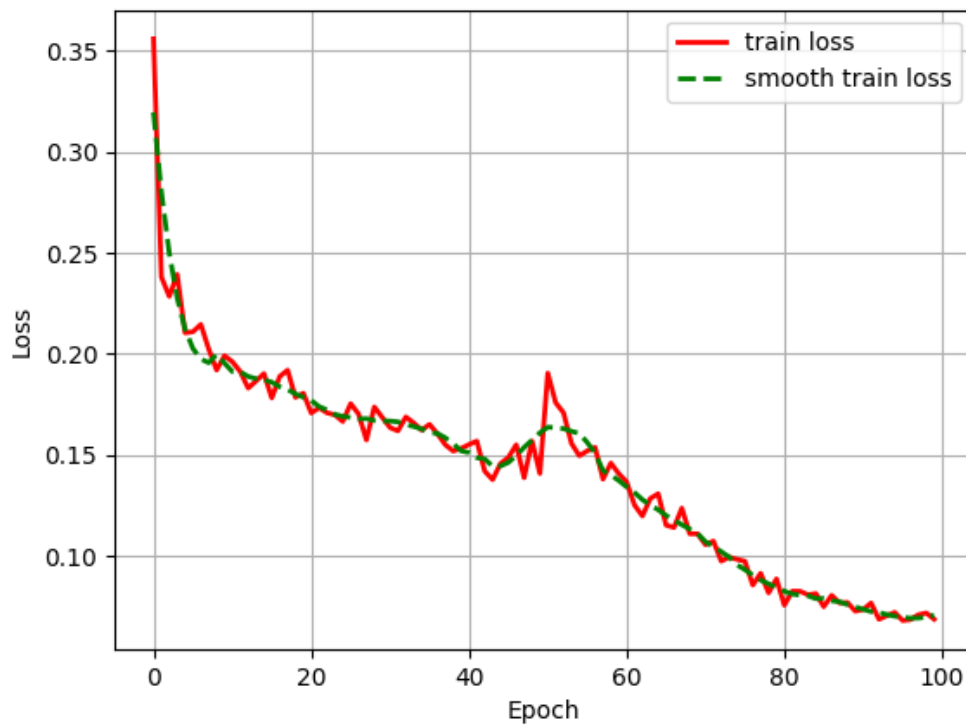
训练得到 22 个模型文件,其中一个 best\_epoch\_weights.pth 为所有模型中损失率最小的一个,一个 last\_epoch\_weights.pth 为最新的模型,其余二十个为训练 100 轮,每五轮保存一次的结果

```

BUSI-256
├── loss_2025_04_18_21_06_38
│   ├── epoch_loss.png
│   ├── epoch_loss.txt
│   ├── events.out.tfevents.1744981598.c
│   ├── best_epoch_weights.pth
│   ├── ep005-loss0.210.pth
│   ├── ep010-loss0.199.pth
│   ├── ep015-loss0.190.pth
│   ├── ep020-loss0.181.pth
│   ├── ep025-loss0.167.pth
│   ├── ep030-loss0.169.pth
│   ├── ep035-loss0.162.pth
│   ├── ep040-loss0.153.pth
│   ├── ep045-loss0.146.pth
│   ├── ep050-loss0.141.pth
│   ├── ep055-loss0.150.pth
│   ├── ep060-loss0.141.pth
│   ├── ep065-loss0.131.pth
│   ├── ep070-loss0.111.pth
│   ├── ep075-loss0.098.pth
│   ├── ep080-loss0.089.pth
│   ├── ep085-loss0.081.pth
│   ├── ep090-loss0.073.pth
│   ├── ep095-loss0.072.pth
│   ├── ep100-loss0.069.pth
│   └── last_epoch_weights.pth

```

从 epoch\_loss.png 即训练轮次-损失率图像可以看出损失率随着训练轮次的变化



从训练日志可以看出 epoch=96 时损失率最小,从 epoch\_loss.txt 可以看出此时 loss=0.06803704285953036

```
Epoch 96/100:  0%|          | 0/283 [00:00<?, ?it/s<class  
'dict'>]Start Train  
Epoch 96/100: 100%|██████████| 283/283 [01:44<00:00,  2.71it/s,  
f_score=0.907, lr=1e-6, total_loss=0.068]  
Epoch:96/100  
Total Loss: 0.068  
Save best model to best_epoch_weights.pth
```

## 预测步骤

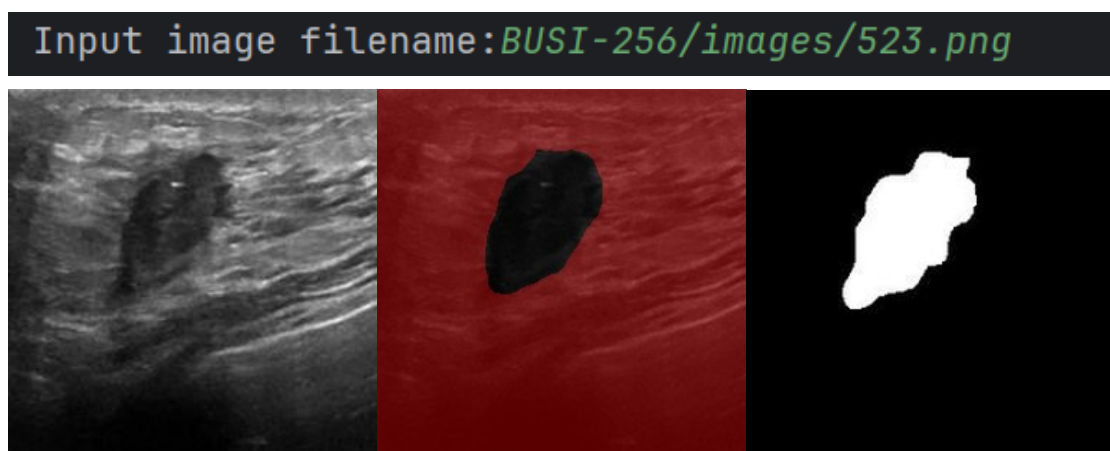
修改 unet.py 文件,将 model\_path 修改为训练好的模型存放位置,num\_classes 修改为 train\_medical.py 中 num\_classes 的数值 input\_shape 修改为输入图片的大小

```
30 "model_path" : 'logs/best_epoch_weights.pth',
31 #-----#
32 # 所需要区分的类的个数+1
33 #-----#
34 "num_classes" : 2,|
35 #-----#
36 # 所使用的主干网络: vgg、resnet50
37 #-----#
38 "backbone" : "vgg",
39 #-----#
40 # 输入图片的大小
41 #-----#
42 "input_shape" : [256, 256],
43 #-----#
```

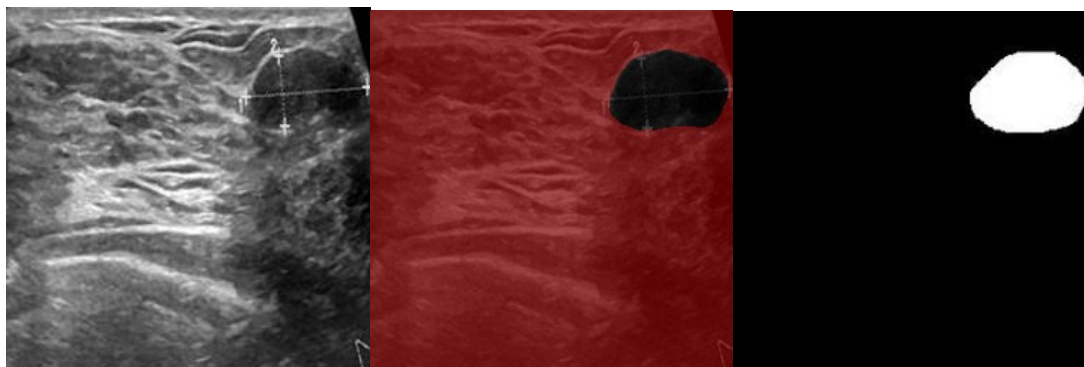
运行 predict.py 开始预测

```
↑ D:\anaconda3\envs\unet\python.exe "D:\Visual studio\code\pycharm\AIOT_UNET\predict.py"
↓ logs/best_epoch_weights.pth model, and classes loaded.
↺ Configurations:
↻ -----#
| keys | values|
|-----#|
| model_path | logs/best_epoch_weights.pth|
| num_classes | 2|
| backbone | vgg|
| input_shape | [256, 256]|
| mix_type | 0|
| cuda | True|
|-----#|
Input image filename:|
```

输入需要预测文件的相对路径



Input image filename: *BUSI-256/images/126.png*

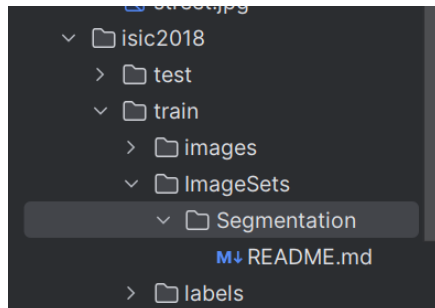


左为原图,中为预测图,右为标签图

## isic2018 数据集

### 训练步骤

#### 文件结构



修改 voc\_annotation.py 并运行

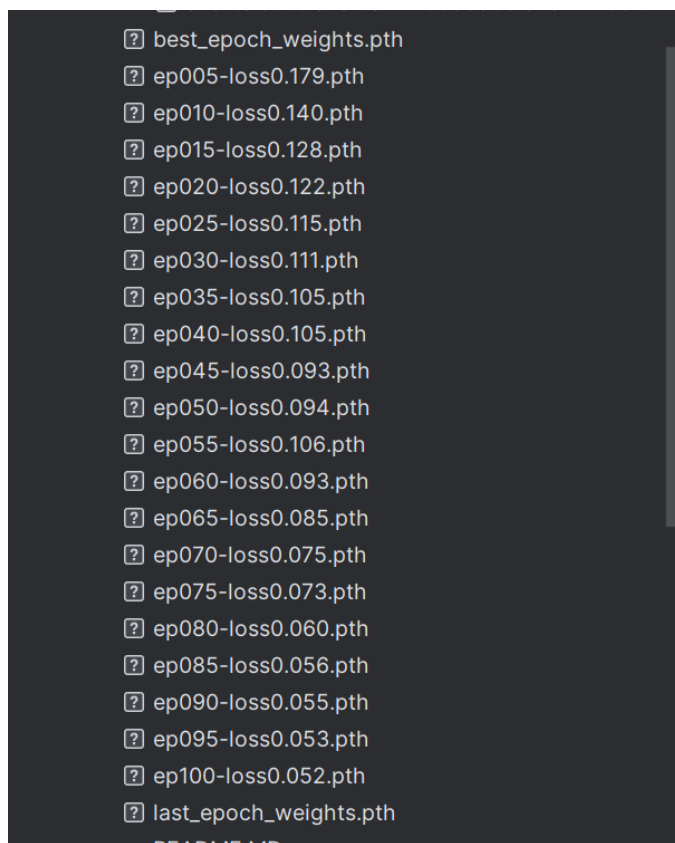
```
VOCdevkit_path = 'isic2018/train' #'BUSI-256' #'Medical_Datasets'
```

修改 train\_medical.py 并运行

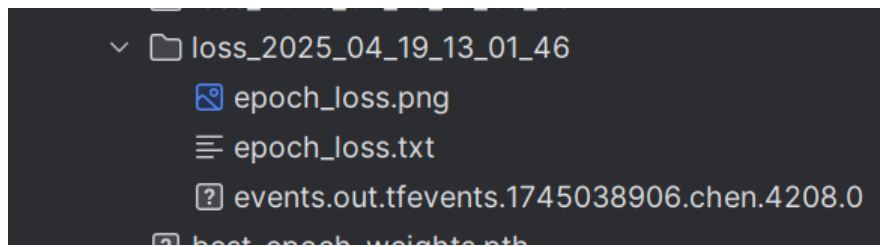
```
116         input_shape = [256, 256]
```

```
VOCdevkit_path = 'isic2018/train' #'BUSI-256' #'Medical_Datasets'
```

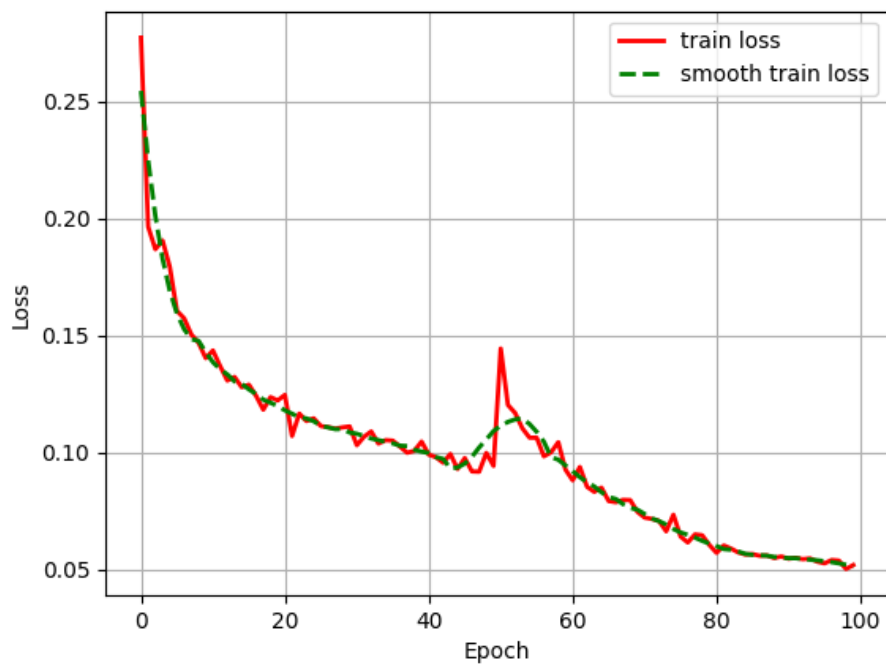
训练得到 22 个模型文件,其中一个 best\_epoch\_weights.pth 为所有模型中损失率最小的一个,一个 last\_epoch\_weights.pth 为最新的模型,其余二十个为训练 100 轮,每五轮保存一次的结果



损失率和训练轮次关系



从 epoch\_loss.png 即训练轮次-损失率图像可以看出损失率随着训练轮次的变化



模型训练完成,在 epoch=99 时获得最佳的损失率,loss= 0.05015246711561727

```
Start Train
Epoch 99/100: 100%|██████████| 943/943 [01:46<00:00, 8.85it/s,
f_score=0.956, lr=1e-6, total_loss=0.0502]
Epoch:99/100
Total Loss: 0.050
Save best model to best_epoch_weights.pth
```

## 预测步骤

修改 unet.py

```
29 ##### "model_path" : 'model_data/unet_vgg_voc.pth',
30 "model_path" : 'logs/best_epoch_weights.pth',
31 #-----#
32 # 所需要区分的类的个数+1
33 #-----#
34 "num_classes" : 2,
35 #-----#
36 # 所使用的的主干网络: vgg、resnet50
37 #-----#
38 "backbone" : "vgg",
39 #-----#
40 # 输入图片的大小
41 #-----#
42 "input_shape" : [256, 256],
43 #-----#
44 # mix_type参数用于控制检测结果的可视化方式
45 #
46 # mix_type = 0的时候代表原图与生成的图进行混合
47 # mix_type = 1的时候代表仅保留生成的图
48 # mix_type = 2的时候代表仅扣去背景, 仅保留原图中的目标
49 #-----#
50 "mix_type" : 0,
51 #-----#
```

运行 predict.py 开始预测

```
D:\anaconda3\envs\unet\python.exe "D:\Visual studio\code\pycharm\AIOT_UNET\predict.py"
logs/best_epoch_weights.pth model, and classes loaded.
Configurations:
-----
|           keys |           values|
-----
|   model_path | logs/best_epoch_weights.pth|
| num_classes | 2|
|   backbone | vgg|
| input_shape | [256, 256]|
|   mix_type | 0|
|      cuda | True|
-----
Input image filename:
```



输入测试数据集的图像相对路径

```
Input image filename:isic2018/test/images/5.png
```



```
Input image filename:isic2018/test/images/356.png
```



左为原图,中为预测图,右为标签图

## 批量预测

新建 predict\_all.py 用于预测 isiu2018 测试集下的所有图像

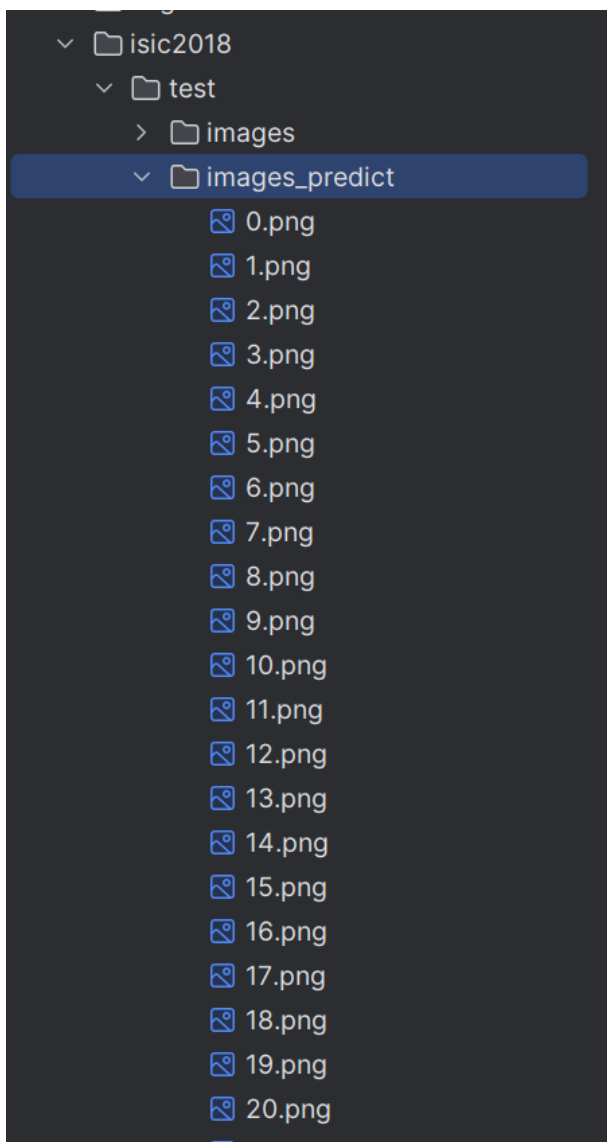
```
import shutil

from PIL import Image
import os

from unet import Unet
if __name__=='__main__':
    mode = "predict_all"
    image_dir='isic2018/test/images/'
    save_dir='isic2018/test/images_predict/'
    if os.path.exists(save_dir):
        shutil.rmtree(save_dir)
    os.mkdir(save_dir)
    unet=Unet()
    dir_list=os.listdir(image_dir)
```

```
for img in dir_list:
    try:
        image = Image.open(image_dir+img)
    except:
        print('Open Error! Try again!')
        continue
    else:
        r_image = unet.detect_image(image)
        r_image.save(save_dir+img)
```

image\_dir 为测试集 test 图像的相对位置,save\_dir 为预测图像的保存位置,可以看到预测图像已成功保存



## 备注

医学图像不可分类,故无法进行计算 miou 等性能指标,只能用于观看数据集的训练效果

BUSI-256 只有训练集,故预测借用训练集,isis2018 train 训练集用于训练模型,test 测试集用于预测,预测图像保存在 isic2018/test/images\_predict 目录下

两个数据集训练出来的模型及数据保存在各自子目录'目录名+train'下,训练过程信息保存在该子目录下的 log.txt 文件中

更新了需求文档 requirement.txt